

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение
высшего образования
«Северный (Арктический) федеральный университет имени М.В. Ломоносова»

Высшая школа информационных технологий и автоматизированных систем

ЛАБОРАТОРНАЯ РАБОТА №1

По дисциплине: Защита информации в системах управления базами данных

На тему Системы хранения данных

Выполнил обучающийся:

Грозов Илья Владимирович

Направление подготовки / специальность:

10.03.01 Информационная безопасность

Курс: 3

Группа: 151113

Руководитель: Зубарев Александр Андреевич, ст.

преподаватель

Отметка о зачете

Руководитель

А.А. Зубарев.

Архангельск 2023

ЗАДАНИЕ

Получить практический навык PostgreSQL Authentication

ХОД РАБОТЫ

1. БАЗОВЫЕ НАСТРОЙКИ DOCKER.IO

1.1 Настройка и установка docker.io

Для установки docker.io необходимо воспользоваться командой: `sudo apt install -y docker.io`. Процесс установки отображен на рисунке 1.1.1

```
(kali@kali)-[~]
└─$ sudo apt install -y docker.io
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  cgroupfs-mount containerd libintl-perl libintl-xs-perl libmodule-find-perl libmodule-scandeps-perl libproc-processtable-perl libsort-naturally-perl needrestart runc tini
Suggested packages:
  containernetworking-plugins docker-doc aufs-tools btrfs-progs debootstrap rinse rootlesskit xfsprogs zfs-fuse | zfsutils-linux
Recommended packages:
  criu
The following NEW packages will be installed:
  cgroupfs-mount containerd docker.io libintl-perl libintl-xs-perl libmodule-find-perl libmodule-scandeps-perl libproc-processtable-perl libsort-naturally-perl needrestart runc tini
0 upgraded, 12 newly installed, 0 to remove and 352 not upgraded.
Need to get 67.8 MB of archives.
After this operation, 272 MB of additional disk space will be used.
Get:2 http://http.kali.org/kali kali-rolling/main amd64 containerd amd64 1.6.24-ds1-1 [26.9 MB]
Get:1 http://http.kali.org/kali kali-rolling/main amd64 runc amd64 1.1.12+ds1-1 [2,757 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 tini amd64 0.19.0-1 [255 kB]
Get:4 http://http.kali.org/kali kali-rolling/main amd64 docker.io amd64 20.10.25+dfsg1-2+b3 [37.0 MB]
Get:5 http://kali.download/kali kali-rolling/main amd64 cgroupfs-mount all 1.4 [6,276 B]
Get:6 http://kali.download/kali kali-rolling/main amd64 libintl-perl all 1.33-1 [720 kB]
Get:7 http://http.kali.org/kali kali-rolling/main amd64 libintl-xs-perl amd64 1.33-1+b1 [15.5 kB]
Get:8 http://kali.download/kali kali-rolling/main amd64 libmodule-find-perl all 0.16-2 [10.6 kB]
Get:9 http://kali.download/kali kali-rolling/main amd64 libmodule-scandeps-perl all 1.35-1 [43.7 kB]
Get:10 http://http.kali.org/kali kali-rolling/main amd64 libproc-processtable-perl amd64 0.636-1+b1 [42.2 kB]
Get:11 http://kali.download/kali kali-rolling/main amd64 libsort-naturally-perl all 1.03-4 [13.1 kB]
Get:12 http://kali.download/kali kali-rolling/main amd64 needrestart all 3.6-7 [59.7 kB]
Fetched 67.8 MB in 3s (25.1 MB/s)
Selecting previously unselected package runc.
(Reading database ... 404978 files and directories currently installed.)
Preparing to unpack .../00-runc_1.1.12+ds1-1_amd64.deb ...
Unpacking runc (1.1.12+ds1-1) ...
Selecting previously unselected package containerd.
Preparing to unpack .../01-containerd_1.6.24-ds1-1_amd64.deb ...
Unpacking containerd (1.6.24-ds1-1) ...
Selecting previously unselected package tini.
Preparing to unpack .../02-tini_0.19.0-1_amd64.deb ...
Unpacking tini (0.19.0-1) ...
Selecting previously unselected package docker.io.
Preparing to unpack .../03-docker.io_20.10.25+dfsg1-2+b3_amd64.deb ...
Unpacking docker.io (20.10.25+dfsg1-2+b3) ...
Selecting previously unselected package cgroupfs-mount.
Preparing to unpack .../04-cgroupfs-mount_1.4_all.deb ...
Unpacking cgroupfs-mount (1.4) ...
Selecting previously unselected package libintl-perl.
Preparing to unpack .../05-libintl-perl_1.33-1_all.deb ...
Unpacking libintl-perl (1.33-1) ...
Selecting previously unselected package libintl-xs-perl.
Preparing to unpack .../06-libintl-xs-perl_1.33-1+b1_amd64.deb ...
Unpacking libintl-xs-perl (1.33-1+b1) ...
Selecting previously unselected package libmodule-find-perl.
Preparing to unpack .../07-libmodule-find-perl_0.16-2_all.deb ...
Unpacking libmodule-find-perl (0.16-2) ...
```

Рисунок 1.1.1 – Процесс установки docker.io

Запустим системный процесс командой: `sudo systemctl enable docker --now`. Запуск системного процесса приведен на рисунке 1.1.2

```
(kali@kali)-[~]
└─$ sudo systemctl enable docker --now
Synchronizing state of docker.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable docker

(kali@kali)-[~]
└─$
```

Рисунок 1.1.2 – Запуск системного процесса

Проверим, что системный процесс запущен при помощи команды: `systemctl status docker`. Проверка статуса системного процесса приведена на рисунке 1.1.3

```
(kali@kali)-[~]
$ systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sat 2024-03-09 02:26:43 EST; 3min 11s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
      Main PID: 93755 (dockerd)
         Tasks: 9
        Memory: 32.6M (peak: 32.9M)
           CPU: 720ms
      CGroup: /system.slice/docker.service
             └─93755 /usr/sbin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Mar 09 02:26:41 kali dockerd[93755]: time="2024-03-09T02:26:41.228588652-05:00" level=info msg="[core] Channel Connectivity change to CONNECTING" module=grpc
Mar 09 02:26:41 kali dockerd[93755]: time="2024-03-09T02:26:41.229280220-05:00" level=info msg="[core] Subchannel Connectivity change to READY" module=grpc
Mar 09 02:26:41 kali dockerd[93755]: time="2024-03-09T02:26:41.229299361-05:00" level=info msg="[core] Channel Connectivity change to READY" module=grpc
Mar 09 02:26:42 kali dockerd[93755]: time="2024-03-09T02:26:42.42925928-05:00" level=info msg="Loading containers: start."
Mar 09 02:26:43 kali dockerd[93755]: time="2024-03-09T02:26:43.167479087-05:00" level=info msg="Default bridge (docker0) is assigned with an IP address 172.17.0.0/16. Daemon option --bip can be used to set a preferred IP address"
Mar 09 02:26:43 kali dockerd[93755]: time="2024-03-09T02:26:43.369152233-05:00" level=info msg="Loading containers: done."
Mar 09 02:26:43 kali dockerd[93755]: time="2024-03-09T02:26:43.469062155-05:00" level=info msg="Docker daemon commit-5d4f983c graphdriver(s)-overlay2 version=20.10.25+dfsg1
Mar 09 02:26:43 kali dockerd[93755]: time="2024-03-09T02:26:43.469232787-05:00" level=info msg="Daemon has completed initialization"
Mar 09 02:26:43 kali systemd[1]: Started docker.service - Docker Application Container Engine.
Mar 09 02:26:43 kali dockerd[93755]: time="2024-03-09T02:26:43.523145188-05:00" level=info msg="API listen on /run/docker.sock"
```

Рисунок 1.1.3 – Проверка статуса системного процесса

Установим `docker-compose` при помощи команды: `sudo apt install docker-compose`. Установка `docker-compose` приведена на рисунке 1.1.4.

```
(kali@kali)-[~]
$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-compose python3-docker python3-dockerpty python3-texttable
The following NEW packages will be installed:
  docker-compose python3-compose python3-docker python3-dockerpty python3-texttable
0 upgraded, 5 newly installed, 0 to remove and 352 not upgraded.
Need to get 270 kB of archives.
After this operation, 1,232 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://kali.download/kali kali-rolling/main amd64 python3-docker all 5.0.3-1 [90.2 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 python3-dockerpty all 0.4.1-5 [11.2 kB]
Get:3 http://kali.download/kali kali-rolling/main amd64 python3-texttable all 1.6.7-1 [11.9 kB]
Get:4 http://kali.download/kali kali-rolling/main amd64 python3-compose all 1.29.2-6 [113 kB]
Get:5 http://kali.download/kali kali-rolling/main amd64 docker-compose all 1.29.2-6 [44.2 kB]
Fetched 270 kB in 1s (400 kB/s)
```

Рисунок 1.1.4 – Установка `docker-compose`

Во время установки будет выведено окно с указанием сервисов, которые будут перезапущены. Сервисы можно изменить при необходимости. Окно с указанием сервисов приведено на рисунке 1.1.5

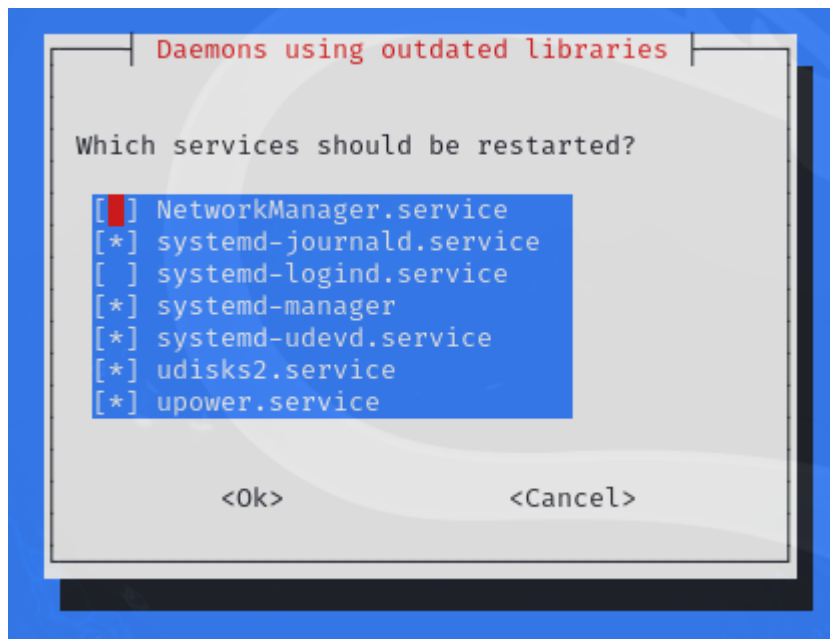


Рисунок 1.1.5 – Окно с указанием сервисов

Проверим установленную версию docker-compose при помощи команды: `docker-compose version`. Проверка установленной версии отображена на рисунке 1.1.6

```
(kali㉿kali)-[~]
└─$ docker-compose version
docker-compose version 1.29.2, build unknown
docker-py version: 5.0.3
CPython version: 3.11.8
OpenSSL version: OpenSSL 3.1.5 30 Jan 2024

(kali㉿kali)-[~]
└─$
```

Рисунок 1.1.6 – Проверка установленной версии

Проверим, что postgresql установлен в системе изначально при помощи команды: `psql --version`

```
(kali㉿kali)-[~]
└─$ psql --version
psql (PostgreSQL) 16.1 (Debian 16.1-1+b1)
```

Рисунок 1.1.7 – Проверка установки postgresql в системе

1.2 Работа с docker-compose

Создадим файл `docker-compose.yml` со следующим содержанием согласно заданию. Содержание файла `docker-compose` приведено на рисунке 1.2.1

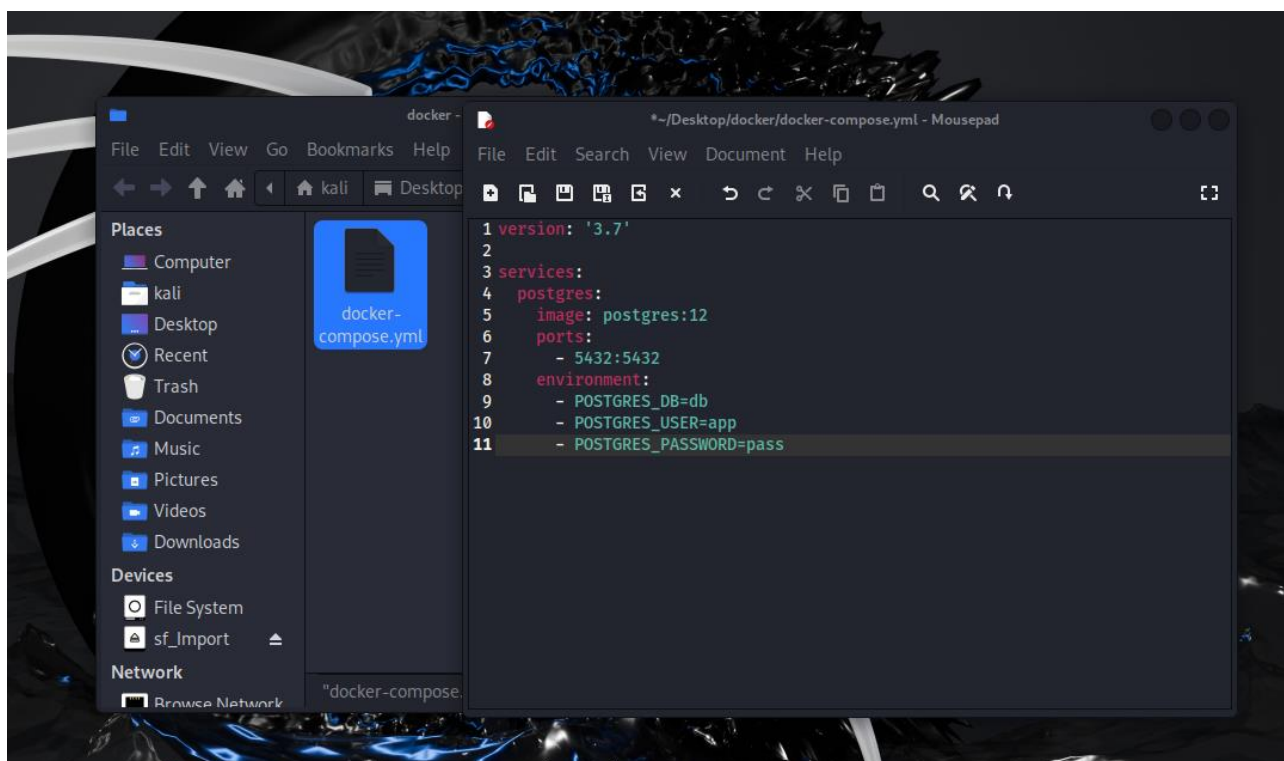


Рисунок 1.2.1 – Содержание файла `docker-compose`

Во время работы `доксер` было запрошено создание сети. Сеть создана с именем `lab1`. Процесс создания сети отображен на рисунке 1.2.2.

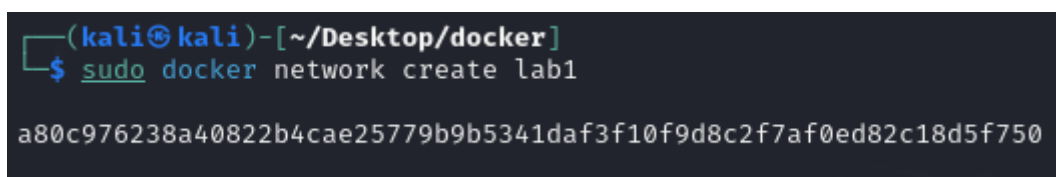


Рисунок 1.2.2 – Процесс создания сети

Перейдем в каталог где был создан наш файл `docker-compose.yml` при помощи команды `cd`. Запустим файл `docker-compose.yml` при помощи следующей команды: `sudo docker-compose up`. Запуск файла `docker-compose.yml` приведен на рисунке 1.2.3.

```
(kali@kali)-[~/Desktop/docker]
$ sudo docker-compose up
Pulling postgres (postgres:12)...
12: Pulling from library/postgres
e1caac4eb9d2: Pull complete
d615c5a045a4: Pull complete
11277ab50e99: Pull complete
17e6a13eda47: Pull complete
e0415688003b: Pull complete
f3bdc5b5936d: Pull complete
42acbddd37ae0: Pull complete
3cba3b5ccdc1: Pull complete
bf6fc39ef2cc: Extracting [=====] 73.53MB/104MB
2eb4ec4e8798: Download complete
9304b4da8753: Download complete
28d509a56ae9: Download complete
81ad19522579: Download complete
b08e3b310533: Download complete
```

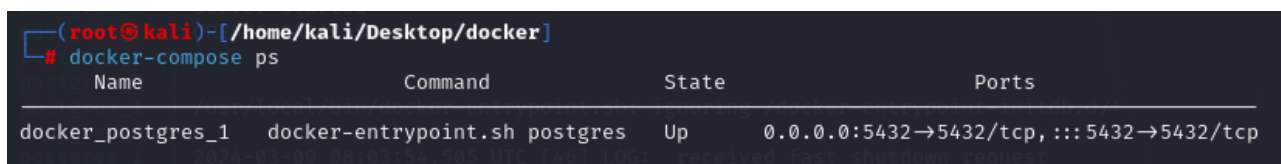
Рисунок 1.2.3 – Запуск файла dokser-compose.yml

По итогу запуска файла в терминале будет выведено следующее содержание отображенное на рисунке 1.2.4. В данном случае запуск произошел успешно. Неудачный запуск может быть вызван нарушением структуры файла dokser-compose.yml или выполнение запуска от имени обычного пользователя.

```
postgres_1 | The database cluster will be initialized with locale "en_US.utf8".
postgres_1 | The default database encoding has accordingly been set to "UTF8".
postgres_1 | The default text search configuration will be set to "english".
postgres_1 |
postgres_1 | Data page checksums are disabled.
postgres_1 |
postgres_1 | fixing permissions on existing directory /var/lib/postgresql/data ... ok
postgres_1 | creating subdirectories ... ok
postgres_1 | selecting dynamic shared memory implementation ... posix
postgres_1 | selecting default max_connections ... 100
postgres_1 | selecting default shared_buffers ... 128MB
postgres_1 | selecting default time zone ... Etc/UTC
postgres_1 | creating configuration files ... ok
postgres_1 | running bootstrap script ... ok
postgres_1 | performing post-bootstrap initialization ... ok
postgres_1 | syncing data to disk ... ok
postgres_1 |
postgres_1 | Success. You can now start the database server using:
postgres_1 |
postgres_1 |     pg_ctl -D /var/lib/postgresql/data -l logfile start
postgres_1 |
postgres_1 | initdb: warning: enabling "trust" authentication for local connections
postgres_1 | You can change this by editing pg_hba.conf or using the option -A, or
postgres_1 | --auth-local and --auth-host, the next time you run initdb.
postgres_1 | waiting for server to start...2024-03-09 08:03:53.943 UTC [46] LOG: starting PostgreSQL 12.18 (Debian 12.18-1.pgdg120+2) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
postgres_1 | 2024-03-09 08:03:53.948 UTC [47] LOG: database system was shut down at 2024-03-09 08:03:52 UTC
postgres_1 | 2024-03-09 08:03:53.977 UTC [46] LOG: database system is ready to accept connections
postgres_1 | done
postgres_1 | server started
postgres_1 | CREATE DATABASE
postgres_1 |
postgres_1 | /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
postgres_1 |
postgres_1 | 2024-03-09 08:03:54.505 UTC [46] LOG: received fast shutdown request
postgres_1 | waiting for server to shut down...2024-03-09 08:03:54.509 UTC [46] LOG: aborting any active transactions
postgres_1 | 2024-03-09 08:03:54.512 UTC [46] LOG: background worker "logical replication launcher" (PID 53) exited with exit code 1
postgres_1 | 2024-03-09 08:03:54.513 UTC [48] LOG: shutting down
postgres_1 | 2024-03-09 08:03:54.542 UTC [46] LOG: database system is shut down
postgres_1 | done
postgres_1 | server stopped
postgres_1 |
postgres_1 | PostgreSQL init process complete; ready for start up.
postgres_1 |
postgres_1 | 2024-03-09 08:03:54.626 UTC [1] LOG: starting PostgreSQL 12.18 (Debian 12.18-1.pgdg120+2) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
postgres_1 | 2024-03-09 08:03:54.626 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
postgres_1 | 2024-03-09 08:03:54.626 UTC [1] LOG: listening on IPv6 address ":::", port 5432
postgres_1 | 2024-03-09 08:03:54.634 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgres_1 | 2024-03-09 08:03:54.659 UTC [75] LOG: database system was shut down at 2024-03-09 08:03:54 UTC
postgres_1 | 2024-03-09 08:03:54.667 UTC [1] LOG: database system is ready to accept connections
```

Рисунок 1.2.4 – Итог запуска файла dokser-compose.yml

Проверим, что контейнер находится в рабочем состоянии при помощи команды: `docker-compose ps`. Выполнение команды осуществляется от имени администратора из каталога где находится файл `docker-compose.yml`. Проверка состояния контейнера отображена на рисунке 1.2.5.



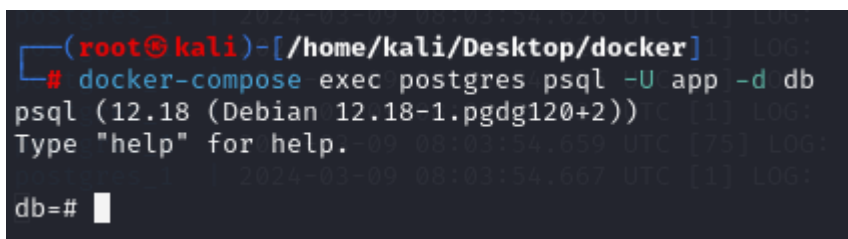
```
(root@kali)-[/home/kali/Desktop/docker]
# docker-compose ps
```

Name	Command	State	Ports
docker_postgres_1	docker-entrypoint.sh postgres	Up	0.0.0.0:5432→5432/tcp, :::5432→5432/tcp

Рисунок 1.2.5 – Проверка состояния контейнера.

1.3 Работы с базой данных

Удостоверимся, что у нас есть доступ к базе данных при помощи следующей команды: `docker-compose exec postgres psql -U -d db`. Подключение к базе данных произошло успешно. Подключение к базе данных отображено на рисунке 1.3.1

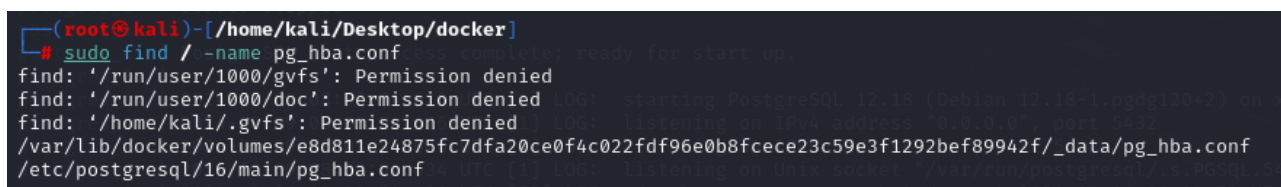


```
(root@kali)-[/home/kali/Desktop/docker]
# docker-compose exec postgres psql -U app -d db
psql (12.18 (Debian 12.18-1.pgdg120+2))
Type "help" for help.

db=#
```

Рисунок 1.3.1 – Подключение к базе данных

Произведем поиск необходимого файла в системе, так как файл по расположению из задания отсутствует. Поиск произведен при помощи команды `sudo find / -name pg_hba.conf`. Поиск файла отображен на рисунке 1.3.2.



```
(root@kali)-[/home/kali/Desktop/docker]
# sudo find / -name pg_hba.conf
find: '/run/user/1000/gvfs': Permission denied
find: '/run/user/1000/doc': Permission denied
find: '/home/kali/.gvfs': Permission denied
/var/lib/docker/volumes/e8d811e24875fc7dfa20ce0f4c022fdf96e0b8fcede23c59e3f1292bef89942f/_data/pg_hba.conf
/etc/postgresql/16/main/pg_hba.conf
```

Рисунок 1.3.2 – Поиск файла

Просмотрим содержимое найденного файла при помощи команды `nano`.
Содержимое файла отображено на рисунке 1.3.3.



```
GNU nano 7.2 /etc/postgresql/16/main/pg_hba.conf
PostgreSQL Client Authentication Configuration File
#
# Refer to the "Client Authentication" section in the PostgreSQL
# documentation for a complete description of this file. A short
# synopsis follows.
#
# Authentication Records
#
# This file controls: which hosts are allowed to connect, how clients
# are authenticated, which PostgreSQL user names they can use, which
# databases they can access. Records take one of these forms:
#
# local      DATABASE USER METHOD [OPTIONS]
# host       DATABASE USER ADDRESS METHOD [OPTIONS]
# hostssl    DATABASE USER ADDRESS METHOD [OPTIONS]
# hostnossl  DATABASE USER ADDRESS METHOD [OPTIONS]
# hostgssenc DATABASE USER ADDRESS METHOD [OPTIONS]
# hostgssenc DATABASE USER ADDRESS METHOD [OPTIONS]
#
# (The uppercase items must be replaced by actual values.)
#
# The first field is the connection type:
# - "local" is a Unix-domain socket
# - "host" is a TCP/IP socket (encrypted or not)
# - "hostssl" is a TCP/IP socket that is SSL-encrypted
# - "hostnossl" is a TCP/IP socket that is not SSL-encrypted
# - "hostgssenc" is a TCP/IP socket that is GSSAPI-encrypted
# - "hostgssenc" is a TCP/IP socket that is not GSSAPI-encrypted
#
# DATABASE can be "all", "sameuser", "samerole", "replication", a
# database name, a regular expression (if it starts with a slash (/))
# or a comma-separated list thereof. The "all" keyword does not match
# "replication". Access to replication must be enabled in a separate
# record (see example below).
#
# USER can be "all", a user name, a group name prefixed with "+", a
# regular expression (if it starts with a slash (/)) or a comma-separated
# list thereof. In both the DATABASE and USER fields you can also write
# a file name prefixed with "@" to include names from a separate file.
#
# ADDRESS specifies the set of hosts the record matches. It can be a
# host name, or it is made up of an IP address and a CIDR mask that is
# an integer (between 0 and 32 (IPv4) or 128 (IPv6) inclusive) that
# specifies the number of significant bits in the mask. A host name
# that starts with a dot (.) matches a suffix of the actual host name.
# Alternatively, you can write an IP address and netmask in separate
# columns to specify the set of hosts. Instead of a CIDR-address, you
#
# Help      Write Out  Where Is  Cut      Execute  Location  Read 132 lines  Set Mark  To Bracket  Previous  Back  Prev Word  Home
# Exit      Read File  Replace  Paste    Justify  Go To Line  Undo      Copy      Where Was  Next     Forward Next Word End
```

Рисунок 1.3.3 – Содержимое файла

2. ИЗМЕНЕНИЕ МЕТОДОВ АУТЕНТИФИКАЦИИ.

2.1 Работа с контейнером dokсер.

Остановим контейнер dokсер. Остановка контейнера dokсер приведена на рисунке 2.1.1.

```
postgres_1 /usr/local/bin/docker-entrypoint.sh: ignoring /docker-entrypoint-initdb.d/*
postgres_1
postgres_1
postgres_1 2024-03-09 08:03:54.505 UTC [46] LOG: received fast shutdown request
postgres_1 waiting for server to shut down...2024-03-09 08:03:54.509 UTC [46] LOG: aborting any active transactions
postgres_1 2024-03-09 08:03:54.512 UTC [46] LOG: background worker "logical replication launcher" (PID 53) exited with exit code 1
postgres_1 2024-03-09 08:03:54.513 UTC [48] LOG: shutting down
postgres_1 2024-03-09 08:03:54.542 UTC [46] LOG: database system is shut down
postgres_1 done
postgres_1 server stopped
postgres_1
postgres_1 PostgreSQL init process complete; ready for start up.
postgres_1
postgres_1 2024-03-09 08:03:54.626 UTC [1] LOG: starting PostgreSQL 12.18 (Debian 12.18-1.pgdg120+2) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit
postgres_1 2024-03-09 08:03:54.626 UTC [1] LOG: listening on IPv4 address "0.0.0.0", port 5432
postgres_1 2024-03-09 08:03:54.626 UTC [1] LOG: listening on IPv6 address "::", port 5432
postgres_1 2024-03-09 08:03:54.634 UTC [1] LOG: listening on Unix socket "/var/run/postgresql/.s.PGSQL.5432"
postgres_1 2024-03-09 08:03:54.659 UTC [75] LOG: database system was shut down at 2024-03-09 08:03:54 UTC
postgres_1 2024-03-09 08:03:54.667 UTC [1] LOG: database system is ready to accept connections
^CGracefully stopping... (press Ctrl+C again to force)
Stopping docker_postgres_1 ... done
[root@kali] - [/home/kali/Desktop/docker]
```

Рисунок 2.1.1 – Остановка контейнера dokсер

Удалим контейнер docker при помощи команды `sudo docker-compose rm`. Подтвердим удаление контейнера. Удаление контейнера отображено на рисунке 2.1.2.

```
(root@kali) - [/home/kali/Desktop/docker]
# sudo docker-compose rm
Going to remove docker_postgres_1
Are you sure? [yN] Y
Removing docker_postgres_1 ... done
```

Рисунок 2.1.2 – Удаление контейнера

2.2 Модификация файла Остановка контейнера docker

Изменим метод аутентификации в файле `docker-compose.yml` на `POSTGRES_HOST_AUTH_METHOD=reject`. Изменение аутентификации отображено на рисунке 2.2.1

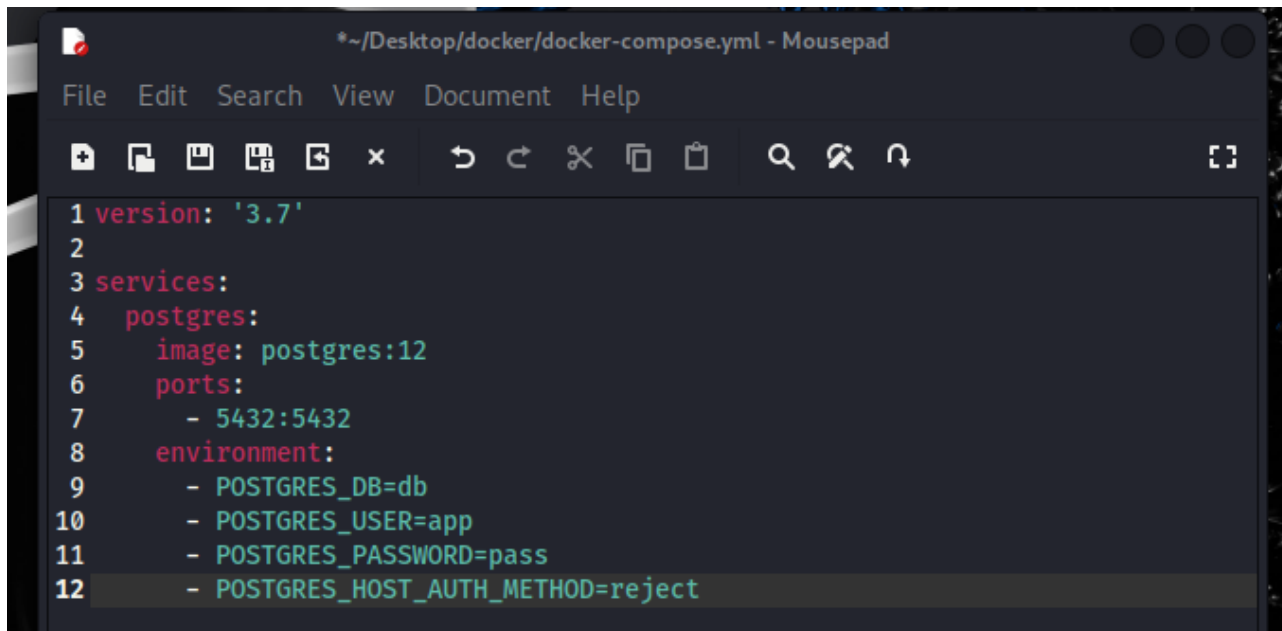


Рисунок 2.2.1 – Изменение метода аутентификации

Запустим измененный контейнер при помощи ранее введенный команды. Повторный запуск контейнера приведен на рисунке 2.2.2.

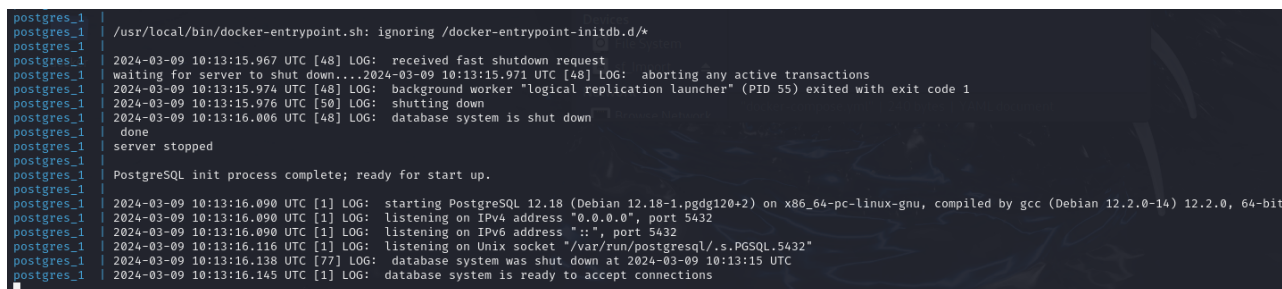


Рисунок 2.2.2 – Повторный запуск контейнера

Выполним подключение к базе при помощи команды: `psql -h localhost -p 5432 -U app -d db -c`. В подключенной базе данных выполним запрос `select * from pg_authid` для получения требуемых значений. Подключение к базе данных и результат запроса приведен на рисунке 2.2.3.

```
(root@kali) ~ | /home/kali/Desktop/docker |
└─ docker-compose exec postgres psql -U app -d db
psql (12.18 (Debian 12.18-1.pgdg120+2))
Type "help" for help.

db=# select * from pg_authid;
 oid |      rolname      | rolsuper | rolinherit | rolcreatorole | rolcreatedb | rolcanlogin | rolreplication | rolbypassrls | rolconnlimit | rolpassword | rolvaliduntil
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
 3373 | pg_monitor        | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 3374 | pg_read_all_settings | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 3375 | pg_read_all_stats  | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 3377 | pg_stat_scan_tables | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 4569 | pg_read_server_files | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 4570 | pg_write_server_files | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 4571 | pg_execute_server_program | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 4280 | pg_signal_backend  | f        | t          | f             | f           | f           | f              | f            | -1           |             |
 10   | app                | t        | t          | t             | t           | t           | t              | t            | -1           | md50d06d8eb2a5f36ab73d83481186543d5 |
(9 rows)

db=#
```

Рисунок 2.2.3 – Подключение к базе данных и результат запроса

КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие методы аутентификации используются для подключения по TCP/IP с адресов 127.0.0.1/32 и ::1/128?

Один из методов "trust" позволяет подключаться к базе данных без запроса пароля, любой пользователь имеющий доступ к системе, может подключиться к базе данных. Метод peer используется для локальных подключений к PostgreSQL, PostgreSQL проверяет операционную систему, сравнивает имя текущего пользователя с именем в базе данных.

2. Какие методы аутентификации используются для подключения по TCP/IP со всех остальных адресов, кроме указанных в предыдущем пункте по протоколу?

Метод password. Он требует предоставить от пользователя базы данных пароль.

3. Верно ли следующее утверждение: пароль роли app хранится в виде функция хеширования password (пароль хранится в поле rolpassword)? Если не верно, то приведите описание алгоритма, который используется для хранения хеша?

Если пароль зашифрован MD5, значение в rolpassword начинается со строки md5, за которой идёт 32-символьный шестнадцатеричный хеш MD5. Этот хеш вычисляется для пароля пользователя с добавленным за ним его именем. Пароль может быть зашифрован и по алгоритму SCRAM-SHA-256. Пароль, не удовлетворяющий ни одному из этих форматов, считается незашифрованным.

4. Какое значение (t будет означать да, f - нет) имеют поля rolsuper, rolcreatorole, rolcreatedb, rolbypassrls с указанием назначения данных столбцов?

Значения полей предоставлены на рисунке К1.

```
(root@kali):~/home/kali/Desktop/docker
$ docker-compose exec postgres psql -U app -d db
psql (12.18 (Debian 12.18-1.pgdg120+2))
Type "help" for help.
```

db=# select *	from pg_authid;										
oid	rolname	rolsuper	rolinherit	rolcreatorole	rolcreatedb	rolcanlogin	rolreplication	rolbypassrls	rolconnlimit	rolpassword	rolvaliduntil
3373	pg_monitor	f	t	f	f	f	f	f	-1		
3374	pg_read_all_settings	f	t	f	f	f	f	f	-1		
3375	pg_read_all_stats	f	t	f	f	f	f	f	-1		
3377	pg_stat_scan_tables	f	t	f	f	f	f	f	-1		
4569	pg_read_server_files	f	t	f	f	f	f	f	-1		
4570	pg_write_server_files	f	t	f	f	f	f	f	-1		
4571	pg_execute_server_program	f	t	f	f	f	f	f	-1		
4200	pg_signal_backend	f	t	f	f	f	f	f	-1		
10	app	t	t	t	t	t	t	t	-1	md50d06d8eb2a5f36ab73d83481186543d5	

```
(9 rows)
db=#
```

Рисунок К1 – Значения полей

5. Почему значения полей `rolcanlogin` и `rolpassword` для роли `app` не изменились и вы по-прежнему можете подключиться с помощью `psql` без указания пароля, хотя в `pg_hba.conf` для `host all all all` указано `reject`?

Подключение к базе данных выполняется ввиду того, что происходит обработка первой записи подходящей по адресу, базе данных и типу соединений. Параметр `reject` в данном случае участия не принимает

ВЫВОД

В ходе выполнения лабораторной работы по теме: «Работа со средствами дизассемблирования и отладки» получили практический навык PostgreSQL Authentication